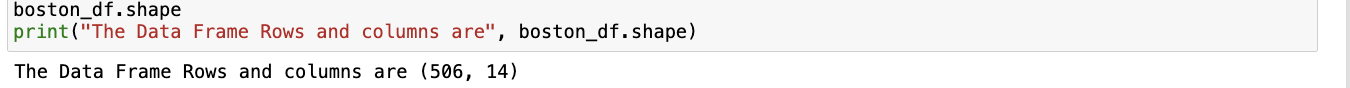
**Case4**

**1)**

**Creating boston\_df**



**Dimensions of Data frame (Number of Rows and Columns)**



**Original Column Names of the Data frame**

A screenshot of a computer

Description automatically generated

**Original Data types of the Columns**

A screenshot of a computer

Description automatically generated

**Changing of Object Datatypes to Dummies**

**A screenshot of a computer code

Description automatically generated**

A screenshot of a computer

Description automatically generated

**Edited Datatypes**

A screenshot of a computer

Description automatically generated

**2 a)**

**Developing Training and Validation Partition (70% Training, 30% Validation)**

A screenshot of a computer

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

The StandardScaler() method to make sure all the data is measured in the same way, even if they originally had different units. This helps to make predictions more accurate and dependable.

**Formula**

|  |
| --- |
| **Zj = (xj – Uj) / Sj** |

Uj = Mean values in Column j.

Sj = Standard Deviation of the numbers in column j.

**2 b)**

**Training of Neural Network Model using MLPRegressor ()**

A screenshot of a computer program

Description automatically generated

**Output**

A screenshot of a computer

Description automatically generated

In the above image for **Intercepts**, we can see that the first array displays the intercepts which are values used in neural networks. It explains that there are two sets of numbers: one for the hidden layer and another for the output layer. Since it's about forecasting, there's only one output.

Coming to **network weights,** the weights of the neural network connecting the input layer to the hidden layer, which includes nine nodes, are shown in the first array. As a result, each predictor variable is connected to a different node in the hidden layer via nine different networks. The second array shows the weights of the neural network connecting the hidden layer to the output layer.

**2 C)**

**Predicted Outcome Variable (First 5 records of Validation)**

A screen shot of a computer

Description automatically generated

**2 D)**

**Accuracy Measures of Training and Validation Partition** A screenshot of a computer

Description automatically generated

The RMSE (Root Mean Squared Error) value for the Training Partition is 1.7272, much lower than the RMSE of the Validation Partition, which is 3.2852. Additionally, the MAPE (Mean Absolute Percentage Error) value for the Validation Partition is higher at 13.5704, compared to 6.6359 for the Training Partition. These differences suggest a possibility of overfitting the data.

Since the Model has low **RMSE and MAPE values** it can be considered for predictions. It is critical to remember to be cautious due to the possibility of a slight chance of overfitting while measuring the performance or accuracy of predictions.

**3 a)**

GridSearchCV () to identify best number of nodes for hidden layer in Boston Housing Neural Network Model

A screenshot of a computer code

Description automatically generated

From the above, we can see that the Improved score of 0.8874 is achieved and the Improved Parameters show the hidden layer sizes as two.

**3 b)**

**Training of Improved Neural Network Using MLPRegressor ()**

A screenshot of a computer program

Description automatically generated

**Final Intercepts and Network Weights of Improved Neural Network Model**

A computer screen shot of a computer

Description automatically generated

we see two arrays. The first array displays the bias values of the two nodes in the hidden layer, while the second array represents the bias value of the single node in the output layer, considering this task involves prediction with only one output.

Regarding the "Improvised Network Weights," the first array showcases the weights of the network connections between the input layer and the hidden layer, which comprises two nodes. This means each predictor connects to a different node in the hidden layer through two separate networks. The second array illustrates the weights of the network connections between the hidden layer and the output layer.

**3 c)**

**Accuracy Measures of Training and Validation Partition**

A screenshot of a computer

Description automatically generated

The training partition's **RMSE** value is **2.6987**, while the validation partition's value is **3.0180**. Coming to the MAPE value, the training partition's **MAPE** value is **10.6354**, while the validation partition's value is **12.1057**. Both the values of RMSE and MAPE are better for the Training Partition as the MAPE and RMSE values of the Validation Partition are higher than that of the Training Partition, which shows that the model might have overfitting. Still, the difference between RMSE and MAPE of Validation and Training Partition is very less and not significant, suggesting that there is no significant overfitting in the model.

Overall, the performance measures imply that there is some overfitting which is not a significant probability of overfitting. Since the overfitting is not significant and not very strong, the neural network model would probably function well enough to make predictions.

**3 D)**

**Comparison of Backward Elimination and Improved Neural Network**

**Accuracy Measures of Validation Set from Backward Elimination Method (From Case-1)**

**A white background with black text

Description automatically generated**

**Accuracy Measures of Validation Set for Improved Neural Network (hidden nodes=2)**

A white background with black text

Description automatically generated

The **RMSE Value** for the **Improved Neural Network** is **3.0180** which is better than that of the **Backward Elimination Method** at **3.7318**, The MAPE value is also much better for the **Improved Neural Network** is **12.1057** is much better than **13.9371** which MAPE value of the Back Elimination Method.

Since both the RMSE and MAPE values are better for the **Improved Neural Network** we conclude that it is the better model among the two.